Patent Application of

David R. Rigney

for

DESCRIPTIVE TITLE OF THE INVENTION:

System, Methods, and Computer Program Product for Analyzing Microarray Data

CROSS REFERENCE TO RELATED APPLICATIONS:

This patent application claims the benefit of U.S. Provisional Application No. 60/227,421, filed August 23, 2000, which is hereby incorporated by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH AND DEVELOPMENT: Not applicable.

REFERENCE TO SEQUENCE LISTING, A TABLE, OR A COMPUTER PROGRAM LISTING APPENDIX: Not applicable.

BACKGROUND--FIELD OF THE INVENTION

The present invention relates generally to the field of bioinformatics, and specifically to systems, methods, and computer program products that make use of digital signal processing, clustering of data, statistical natural language processing, and machine learning, for purposes of analyzing data acquired using DNA arrays that have been hybridized with cDNA probes.

BACKGROUND--DESCRIPTION OF PRIOR ART

Disease processes, as well as physiological responses to agents such as drugs, are often investigated by measuring the amounts of different messenger RNA (mRNA) species in a tissue specimen or in a cultured cell population. The present invention is concerned with analyzing such data, in particular, data acquired through use of a recently developed tool known as microarrays [DUGGAN et al., Nature Genetics 21, Suppl. 1:10-14 (1999)].

Microarrays consist of hundreds or thousands of spots of different DNA sequences, corresponding to many different genes, arranged in a grid pattern on a glass substrate or nylon membrane. Complementary DNA (cDNA) prepared from the mRNA of a tissue specimen is hybridized to the microarray, which is then detected by fluorescence or autoradiographic methods. The signal detected at each of the many spots on the microarray is then used as an indication of the relative amount of the corresponding mRNA species in the specimen. Microarray experiments are often performed to compare mRNA levels from tissues under two conditions (e.g., cancerous vs. normal

cells; before vs. after administration of a drug), in which case, the ratio of estimated mRNA levels for each microarray spot under the two conditions is also ordinarily calculated. The construction or interpretation of such ratio estimates may benefit from the application of statistical corrections, especially when spot values are close to the threshold of measurement detectability [CHEN et al., Patent US 6,245,517 (2001); NEWTON et al., www.stat.wisc.edu/~newton/papers /publications (1999).]

Microarrays have also been used to monitor the time course of mRNA levels in a cell population that had been subjected to an intervention, such as a shift in serum concentration in the growth medium, which alters the concentration of hormones and other factors needed for cell growth [IYER et al., Science 283:83-87 (1999)]. Those microarray measurements are typically made from mRNA collected at short time intervals (on the order of several minutes) immediatedly after application of the intervention, and longer intervals thereafter (hours). cDNA prepared from each of these mRNA samples is ordinarily hybridized to a separate array. Ratios are then constructed for each time point, as mentioned above, by dividing the measurement at the time point by a measurement corresponding to time-zero. After inspecting the time course of estimated mRNA levels for all the genes on the arrays in those experiments, investigators noted that the mRNA levels for certain groups of genes tend to fluctuate up and down together. Subsequently, computer algorithms were used to group together sets of genes (known as "clusters", produced by a clustering algorithm) according to the similarity of the time-course of their estimated mRNA levels, making the groupings more objective and relieving investigators of the burden of grouping the

genes by eye [EISEN et al., Proc. Natl. Acad. Sci. USA 95:14863-14868 (1998); TAVAZOIE et al., Nature Genetics 22:281-285 (1999); TAMAYO et al. Proc. Natl. Acad. Sci. USA 96:2907-2912 (1999); BEN-DOR et al. J. Computational Biol. 6:281-297 (1999), GETZ et al., arXiv:physics/9911038 at xxx.lanl.gov (1999); ZHENG et al., Patent US 6,263,287 (2001)].

Microarrays have also been used to measure cell responses to several different types of interventions, at a single time point, rather than the response to a single intervention at a series of time points. In those experiments, groups of genes were also observed to exhibit similar mRNA levels in response to the various interventions, and groupings of those genes were also produced automatically by using clustering algorithms [PEROU et al., Proc. Natl. Acad. Sci. USA 96:9212-9217 (1999); TIBSHIRANI et al., http://www-stat.stanford.edu/~tibs/lab/publications.html (1999)].

The similarity of estimated mRNA levels -- observed among genes in individual clusters -- could in some instances be coincidental, but most investigators attribute the similarity of mRNA levels to unknown biological control mechanisms, whereby functionally related genes are transcribed in a coordinated fashion in order to participate stoichiometrically in a biochemical or cell-physiological process. Thus, the clustering of genes on the basis of the similarity of their mRNA levels is viewed by investigators as an intial step in identifying functionally significant biochemical pathways or cell-physiological processes and their mechanisms of transcriptional control. For example, genes involved in mediating progression through the cell cycle

may be found in the same cluster [IYER et al., supra]. However, it has also been observed that genes with supposedly similar known functions do not always appear together in the same clusters [TAVAZOIE et al., supra]. This may be due in part to inadequacy of the particular clustering algorithm that was used. If a different clustering algorithm were applied to the data, it would generally produce different clusters and may be more successful at grouping together functionally related genes.

Initially, investigators applied hierarchical clustering algorithms to array data [EISEN et al., supra]. Later investigators used self-organizing maps, to perform clustering [TAMAYO et al., supra]. Other investigators have performed clustering of microarray data using the k-means algorithm [TAVAZOIE et al., supra], a graph theoretical algorithm [BEN-DOR et al., supra], super-parametric clustering [GETZ et al., supra], as well as grid and σ-τ clustering [ZHENG et al., supra]. Variations of these algorithms have also been implemented by using various normalizations and distance measures. Additional clustering algorithms were described for situations in which data are parameterized by two or more variables [TIBSHIRANI et al., supra]. Considering that hundreds of other general-purpose clustering algorithms have been described [KAUFMAN and ROUSSEEUW. Finding Groups in Data: An Introduction to Cluster Analysis, Wiley (1990) and references contained therein], many of which may eventually be applied to microarray data, and considering that all of these clustering algorithms may group microarray data in different ways, investigators have the problem of deciding which of those algorithms is most useful for analyzing their data.

The inability to group functionally related genes into individual clusters may also be due to factors other than the use of a sub-optimal general-purpose clustering algorithm, for the following reason. It is thought that the similarity of mRNA levels for the various genes in each cluster may be due to co-regulation of those genes by shared transcription factors. In fact, some investigators use an algorithm that simultaneously clusters genes on the basis of the similarity of their estimated mRNA levels, as well as whether those genes exhibit shared DNA binding sites to which the transcription factors can bind [HOLMES et al., Proc. Int. Conf. on Intelligent Systems for Molecular Biology 8:202-210 (2000)]. If genes in a cluster are co-regulated by the same transcription factors, it would consequently be more appropriate to cluster genes on the basis of the similarity of their mRNA synthesis rates (which the transcription factors affect directly), rather than total mRNA levels. However, the clustering methods described above have heretofore been applied only to microarray data corresponding to the total amount of mRNA for each gene, which is the net amount of mRNA resulting from each gene's mRNA synthesis, minus the amount of the gene's mRNA that has been degraded. One objective of the present invention is therefore to provide a method for estimating mRNA synthesis rates from measurements corresponding to total mRNA levels, for each of the genes represented on a microarray, for purposes of clustering.

Whether clustering is performed using total mRNA levels or estimated mRNA synthesis rates, one needs to compare results made with different clustering algorithms, in order to decide which algrothm is most useful for the data under

investigation. The comparison may be made first in terms of the statistics of how well members of each cluster resemble their corresponding centroid (i.e., tightness of clustering), or in terms of a figure of merit obtained using a resampling approach [YEUNG et al., http://www.cs.washington.edu/homes/kayee/research.html (2000)]. However, such goodness-of-fit comparisons do not assess the quality of clustering in terms of the biological reasonableness of the results, which must be based on the physiological functions of the genes in the clusters.

However, there is little prior art that can assist investigators in evaluating the extent to which genes in clusters are functionally related, which has been taken to be a primary criterion upon which the quality of clustering is judged. The main difficulty in establishing functional relations among genes in clusters lies in the unavailability or incompleteness of factual databases that explicitly link the known functions of genes with one another. TAVAZOIE et al., supra, indexed yeast genes using the 199 functional categories in the Martinsreid Institute of Sciences functional classification scheme database (ribosomal, mitochondrial, TCA pathway, etc.). For each cluster of genes, they then calculated probabilities (P values) of the frequency of observed genes in the various functional categories, to determine whether particular clusters are significantly composed of genes associated with particular functional categories. However, such functional classification databases are available to characterize the genes of only a limited number of organisms, or they may not contain a complete list of known genes. Furthermore, those databases force genes into a predetermined classification scheme that may contain overly-broad or overly-narrow classifications,

or classifications that are not mutually exclusive. Possibly for this reason, TAVAZOIE et al. supra, found that genes with supposedly similar known functions -- as defined by the Martinsreid Institute of Sciences functional classification scheme database for yeast -- do not preferentially appear together in the same cluster.

Consequently, most investigators simply review the lists of clustered genes manually and then offer expert commentary about the functional significance of genes in the various clusters, based on their reading of the literature about those genes. For example, IYER et al., supra, describe one cluster as being enriched for genes "involved in mediating progression through the cell cycle", describe another cluster as containing genes encoding "proteins involved in cellular signaling", and for other clusters they offer no description. At the present state of the art, expert human judgement may well be the best method for evaluating the relatedness of functions of genes in clusters. However, this method is limited by the expertise of its practitioners, as well as by the considerable labor involved in manually reviewing literature concerning the many genes that may be present in the clusters. In fact, even the task of identifying the relevant articles in the scientific literature is arduous.

It is therefore another objective of the present invention to produce automatically generated, quantitative indices (figures-of-merit) of the extent to which genes in a cluster are functionally related to one another, based on information within scientific literature concerning genes present on a microarray. Investigators may use the indices to evaluate the functional relatedness of genes in clusters that were made using a

particular clustering algorithm, as well as to compare the performance of different clustering algorithms. In so doing, the investigators use the figure-of-merit indices that are generated by the method to evaluate the quality of the clustering algorithms, based solely on the content of the literature about the genes associated with the clusters.

In one embodiment of the invention, the figure-of-merit indices are calculated by obtaining text in the scientific literature about genes on a microarray (using an original method that is part of the invention); by putting that literature in groups defined by microarray clustering of the corresponding genes; and by then constructing a mathematical model of the text. The purpose of the model is to identify words or phrases that are most uniquely associated with the text corresponding to each cluster, and that also best distinguish each cluster from the others. Indices are then generated by testing the model's ability to classify additional text about genes in the clusters. The figureof-merit indices of the method and system relate to the percentage of times that the tested classifications are made correctly, as compared with classifications performed on text corresponding to genes placed randomly in clusters.

An advantage of the present method and system is that it does not presuppose the existence of a structured database of gene annotations, such as the Martinsreid Institute of Sciences functional classification scheme database for yeast, which was mentioned above. A further advantage of the present system is that it automatically generates a list of words or phrases ("annotations") that best describe each cluster and that also best distinguish each cluster from the others. The present method and system

produces those words and phrases in a different manner than what was outlined by SHATKAY et al., Internat. Conf. on Intelligent Systems in Molecular Biology 8:317-323 (2000). Unlike the present invention, their method does not make use of information from the clustering of microarray data, and it provides no figure of merit for the quality of microarray clustering. Furthermore, they use a semi-automatic -- rather than automatic -- method that attempts to find literature citations and keywords that are conceptually related to single documents, which must be specified by the user for each gene. The present method and system also produces words and phrases in a different manner than what was described by MASYS et al., Bioinformatics 17:319-326 (2001). Unlike the present invention, their method does not provide a figure of merit for the quality of microarray clustering. Their method also has the disadvantage that the words and phrases it produces are voluminous and generally non-specific, placing a significant burden of interpretation on the investigator, because it links sets of genes to the published literature by way of keyword hierarchies using the entire set of descriptors contained in MeSH and Enzyme Commission nomenclature.

BRIEF SUMMARY OF THE INVENTION

A system and methods for analyzing microarray data includes a computer having a central processing unit and a computer memory, which are used to run computer program modules. The computer program modules, along with experimental signal data representing relative concentrations of particular mRNA species (indexed as nucleic acid accession numbers), are loaded initially into the computer memory from a computer disk. One computer program module groups the mRNA species (or nucleic acid accession numbers) into clusters, each cluster being a subset of the mRNA species (or of the corresponding nucleic acid accession numbers). An option of this clustering module is to estimate mRNA synthesis rates from total mRNA measurements, then perform the clustering using the mRNA synthesis rates. Other computer program modules associate multiple unique identifiers, corresponding to scientific publications describing gene structure and functions, with each of the genes in each of the clusters.

In one embodiment of the invention, a computer program module obtains literature abstracts and other text corresponding to the above-mentioned literature unique identifers. A computer module organizes that text in computer files according to the clustering of the corresponding genes, then constructs a mathematical model of the text. The purpose of the model is to identify words or phrases in the text that are most uniquely associated with the text corresponding to each cluster, and that also best distinguish each cluster from the others. Thus, the method and system automatically generate words and phrases that characterize the functional or structural or interactional relations among genes within the clusters. Another computer program

module produces a quantitative index of the relatedness of genes within each cluster, by testing the model's ability to classify additional text about genes in the clusters. In this embodiment, a figure-of-merit index, which is generated by the method and system, relates to the percentage of times that the test classifications are made correctly, as compared with classifications performed on text that had been clustered randomly.

In another embodiment of the invention, a computer program module calculates an index of the functional relatedness of genes within each cluster, by calculating the average fraction of times that pairs of genes in the cluster are associated with the same literature unique identifier. Another computer program module randomizes the assignment of genes to clusters, then calculates the percentage of times that the index of functional similarity could have occurred by chance.

Output data are accumulated and presented concerning the above-mentioned clusters, words and phrases, as well as the indices of functional relatedness of genes within clusters.

Objects and Advantages

Objects and advantages of the present invention include the following:

(1) The prior art clusters only data that are obtained directly from the hybridization of cDNA to microarrays, without prior transformation other than that intended to correct for (or censor) noise and other errors introduced by the measurement process, e.g., the subtraction of unwanted background that is present in images of microarrays, or the normalization of different microarray images so that reference spots have the same value in all images. An object of the present invention is to mathematically transform

the error-corrected microarray data in such a way that subsequent clustering will organize genes represented on the microarray into groups, based on mathematical models of the mechanism of the co-regulation of genes in clusters. In particular, the present invention extracts from the error-corrected microarray data estimates of the mRNA synthesis and degradation rates for each gene, which may be clustered (by any general-purpose clustering method) for purposes of explicitly identifying genes that have similar mRNA synthesis rates, which would reflect their induction by the same transcription factors.

(2) The prior art provides no methods for obtaining quantitative indices for judging the quality of microarray clustering results, independent of the microarray data themselves, other than those involving databases that have already clustered genes into predetermined functional classes, e.g., the Martinsreid Institute of Sciences functional classification scheme database for yeast. An objective of the present invention is to generate quantitative indices about the functional, structural, or biochemical pathway relatedness of genes in clusters, using literature databases such as PubMed, in which the connections between genes are implicit in the frequency with which literature about different genes uses the same words and terms. An advantage of such quantitative indices is that, in this invention, they can be generated automatically, and their generation does not force genes into a predetermined classification scheme, which may contain overly-broad or overly-narrow classifications. A further advantage of the invention is that it generates, in a totally automatic manner, a list of key words or terms that not only characterize each cluster but that also distinguish each cluster from all the other clusters. A yet further advantage of the invention is that it provides

an automatic method for identifying the relevant literature for purposes of automatic analysis of the quality of clustering. A still further advantage of the invention is that for genes associated with a cluster, it provides a ranking of the importance of those genes on the basis of the relevance of text in literature about the set of genes in a cluster. A further advantage of the invention is that it ranks the relatedness of a cluster to all the other clusters, on the basis of the similarity of text in literature about genes in the clusters.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a preferred embodiment of the system and computer program product for analyzing microarray data.

Fig. 2 is a block diagram of an alternate embodiment for the system and computer program product for analyzing microarray data.

Fig. 3 is a graph showing output from the system and computer program in Fig. 1, providing an example in which the system finds good evidence that a cluster's members are functionally related to one another (Cluster D in IYER et al., supra).

Fig. 4 is a graph showing output from the system and computer program in Fig. 1, providing an example in which the system finds no evidence that a cluster's members are functionally related to one another (Cluster B in IYER et al., supra).

Fig. 5 is a graph showing output from the system and computer program of Fig. 1, providing an example in which clustering was performed on microarray data described in IYER et al., supra, using the estimated mRNA synthesis rate method, in which the system finds evidence that a cluster's members are functionally related to one another.

Fig. 6 is a graph showing output from the system and computer program of Fig. 1, with data and analysis the same as in Fig. 5, except that artificial noise was added by the system to the microarray data before analysis, where the noise had a coefficient of variation of 35%.

Fig. 7 is a graph of simulated microarray time series data involving 100 microarray spots (genes), consisting of 10 sets (clusters) of 10 genes, with each gene in a set having the same mRNA synthesis function but a different mRNA degradation function.

Fig. 8 is a graph showing the results of applying the system's method for estimating mRNA synthesis rates, applied to the data shown in Fig. 7, revealing the 10 clusters.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention is not limited to any particular hardware or operating system environment. Those skilled in the art will understand that the systems and methods may be implemented using a variety of computer platforms, operating systems, and programming languages. Therefore, the following description of specific embodiments of the present invention is for purposes of illustration only.

Hardware

The device for analyzing microarray data is shown in **Fig. 1**. It consists of a computer workstation (**100**), which has a Pentium III central processor unit, abbreviated as CPU (**102**), and which is connected to the Internet using a cable modem and a 10/100 network interface card (**108**). The workstation's User Interface (**104**) makes use of a monitor, keyboard, and mouse. The workstation's operating system, computer program modules, and data repository are loaded into a 512 Megabyte computer memory (**110**) from files on a 34 Gigabyte hard storage disk (**106**).

Operating System

The operating system for the workstation (**112**) is Microsoft Windows 98, augmented with additional software that allows operation of the workstation to resemble a UNIX system, namely, operate in the GNU software environment, which is described at the Web site *http://www.fsf.org*. This additional software is DJGPP version 2.03 [Delorie (1997)], obtained over the Internet at *ftp://ftp.simtel.net/pub/simtelnet/gnu/djgpp*. The workstation's augmented operating

system contains all the utility programs that are customarily downloaded for installation with DJGPP, such as perl, flex, bash, grep, and GNU Fileutils.

Installation of DJGPP is necessary in order to install the text modeling program module **(126)**, keyword identification module **(128)**, and text classification module **(130)**. These modules make use of the "bow" software toolkit for statistical language modeling, text retrieval, classification, and clustering (version 1999.11.22), which was downloaded from the Web site *http://www.cs.cmu.edu/~mccallum/bow*, and which is hereby incorporated by reference, in particular, its open source code. Installation of the "bow" software toolkit was performed as described in the instructions that come with the toolkit, except that the system utility program "autoconf" was run to produce a new "configure" file, instead of using the "configure" file that came with the distributed "bow" software.

Computer Program Product

The computer program modules **(114)** are loaded from the computer storage disk **(106)** into the computer memory **(110)** in order to perform individual tasks in the analysis of microarray data. Because some Users may already have the hardware and operating system software of the system, which are described above, the computer program modules **(114)** along with the associated data repository **(138)** by themselves constitute a computer program product, which such Users would then install on their available computers.

Process Control and Specifying Options

The analysis process is initiated by the User at the User Interface **(104)**, for example, by typing the name of the Process Control Module **(116)** at a DOS prompt,

which causes the operating system to load and run the Process Control Module **(116)**. At the time of inititiation, the User may also specify analysis options, for example in an "argv[]" command line format that is used for computer programs written in the C Language. The options instruct the Process Control Module to proceed in a manner other than by using default parameters of the analysis process. These options are stored for subsequent use as the values of variables within the Process Control Module.

The Process Control Module, in turn, loads and runs other modules, many of which could actually be run independently by the user. For example, the User could type the following MS-DOS or bash command line to cause a computer program called "rainbow" to perform text classification of text files located in subdirectories of a directory called "clusters_text", and then place the results of the classification in a directory called "text_model" (as described in documentation at the Web site for the "bow" software , *http://www.cs.cmu.edu/~mccallum/bow* [McCALLUM (1998)]):

```
rainbow -d .\text_model --index .\clusters_text\*
```

However, the Process Control Module **(116)** does the same thing automatically as one of the steps in the overall analysis process, by running the Text Modeling Module **(126)** that in turn runs the "rainbow" program. The program line within the Text Modeling Module **(126)** that does so, written in the program language C, is as follows:

```
system("rainbow -d .\\text_model --index .\\clusters_text\\*");
```

It is well known by computer programmers that the same type of process control can also be accomplished by using a shell script, or by jumping directly to computer

memory locations corresponding to compiled libraries or subroutines, rather than by invoking such a system("...") function within a compiled computer program.

Glossary

Description of the method used in the workstation system **(100)** to analyze microarray data follows. In order to assist the reader to understand the description, the following glossary of terms is first provided.

Accession number or gene accession number: A researcher who sequences some DNA often deposits that sequence information into a public DNA sequence database, such as GenBank. When the curators of the DNA sequence database enter that sequence information into the database, they assign a unique, permanent identifier to the sequence, which is known as the sequence's accession number. It consists of an alpha-numeric string, such as "W95909". DNA in each spot on the microarray has been sequenced (in whole or in part), and that sequence may be ascertained from a database like GenBank by specifying the known accession number corresponding to that spot.

UniGene Number: Many of the DNA sequences that have been assigned accession numbers are similar to the sequences corresponding to other accession numbers, because different researchers may have sequenced the same gene and deposited the sequence information independently. UniGene is a database that groups together different accession numbers corresponding to similar sequences. Each such grouping of accession numbers is assigned a number (UniGene number). Thus, the accession numbers that have been assigned a particular UniGene number have DNA sequences that are similar to one another.

LocusLink Number: The Locus Link database groups together entries from several databases that involve not only DNA sequence information (such as UniGene data), but also such things as the chromosomal location of the DNA sequence, the Enzyme Commission number of the corresponding protein (if it is an enzyme), and the diseases with which the corresponding gene are associated (if known). Each grouping of database entries about a particular gene is assigned a number (Locus Link number).

Omim Number: Omim is an abbreviation for "Online Mendelian Inheritance in Man", which is a database that describes the connection between particular genes and diseases. Each gene in the Omim database is assigned an identifying number (Omim number). The Locus Link database entries include the Omim number if it exists. There is a Web page for each of the genes in the Omim database, which contains links to relevant scientific literature about the corresponding gene.

UID or PubMed UID: PubMed is a database of biomedical scientific publications. Every scientific article from the journals that are indexed by PubMed is identified by a unique, permanent number -- the Unique IDentifier number ("UID" or "uid").

rainbow: This is the name of a computer program written by McCallum and colleagues at Carnegie Mellon University. It may be used to classify text files. Given examples of text files that deal with a specified number of different subjects (classes), it first examines the text in those files to determine the vocabulary that distinguishes the different subjects (classes). Then, given a text file on an unknown subject, it classifies that file as pertaining most closely to one of the different known subjects, based on the vocabulary that it contains.

Process of Analyzing Microarray Data

The Process Control Module **(116)** begins the analysis by initiating tasks that are to be accomplished by the Initialization Module **(118)**. The first such task is to read a file of gene accession numbers, corresponding to the DNA species that have been spotted onto the microarray under investigation, for which partial sequences are known. The list of accession numbers are stored for future use in the Data Repository **(138)**, in the section Accession Numbers **(140)**.

The next task of the Initialization Module **(118)** is to associate each of the accession numbers with a gene that has been characterized (if possible). It does so by first converting accession numbers to UniGene numbers. It then converts UniGene numbers to LocusLink numbers. Finally, it converts LocusLink numbers to Omim numbers. Each of these conversions is accomplished by using look-up table data, in which the associations between the labelings of different types of data are listed explicitly. These steps are described in more detail in the paragraphs that follow.

The data pointing from accession numbers to UniGene numbers are read by the Initialization Module **(118)** from a file on the storage disk **(106)** and stored in the Data Repository in the section UniGene Data **(142)**. The data in that file are obtained for each new build of UniGene by downloading the file Hs.data.Z (for human genes) from the Web site *ftp.ncbi.nlm.nih.gov/pub/schuler/Unigene* then extracting the accession-to-unigene number data and formatting the data for storage on the disk **(106)**. The extraction consists of uncompressing the compressed file to become Hs.data, then scanning the resulting text file for lines containing ID Hs."u" followed by SEQUENCE

ACC="a", where "u" is a UniGene number and "a" is a corresponding accession number.

The procedure described above pertains to microarray data concerning human DNA. For non-human species (e.g., mouse and rat), the "Hs" in the paragraph above is replaced by the symbol for each of the corresponding species. If the microarray data under analysis are non-human, an additional step is performed to convert the non-human indices to those for the homologous human genes. This extra step consists of using a look-up table to link the UniGene number for the non-human gene with the homologous human UniGene number. The file hmlg.ftp, downloaded from the Web site *ftp.ncbi.nlm.nih.gov/pub/Homologene* and stored on the workstation's disk **(106)**, contains that look-up table data. It is read by the Initialization Module **(118)**, which stores the data in the UniGene Data section **(142)** of the Data Repository **(138)**.

The data pointing from UniGene to LocusLink numbers are also read by the Initialization Module **(118)** from a file on the storage disk **(106)** and stored in the Data Repository in the section LocusLink Data **(144)**. The data in that file had also been extracted from the file Hs.data. The extraction consists of scanning the text file for lines containing ID Hs."u" followed by LOCUSLINK "L", where "u" is a UniGene number and "L" is the corresponding LocusLink number.

The data pointing from LocusLink to Omim numbers are also read by the Initialization Module **(118)** from a file on the storage disk **(106)** and stored in the Data Repository in the section LocusLink Data **(144)**. The data in that file had been extracted from the file LL_templ, downloaded from the Web site *ftp.ncbi.nlm.nih.gov/refseq/LocusLink* . The extraction consists of scanning the text

file for lines containing >>"L" followed by OMIM:"o", where "L" is the LocusLink number, and "o" is the corresponding Omim number.

The Initialization Module **(118)** then links the Accession Numbers in **(140)** to Omim numbers by following the pointers to UniGene numbers, then to LocusLink numbers, then to Omim numbers. Sometimes, no Omim number can be associated with an accession number because the corresponding pointers do not exist. In that case, the accession number is associated with a number (zero), indicating that no corresponding Omim number could be found. Sometimes, more than one Omim number can be associated with an accession number, for example, because the LocusLink table links a UniGene number with more than one LocusLink number. In that case, all of the corresponding Omim numbers are placed in the table associated with that accession number entry, which is stored in the LocusLink Data section **(144)** in the Data Repository **(138)**.

Upon completion of these steps by the Initialization Module **(118)**, the Process Control Module **(116)** initiates operation of the UID identification module **(120)**. Its function is to construct lists of literature Unique IDentifier ("UID" or "uid") numbers, which uniquely identify publications in the scientific literature that describe the genes associated with the accession numbers of the microarray spots. The UID identification module constructs those lists by first downloading Web pages associated with the Omim numbers, those numbers having been obtained as described in the previous paragraph. The UID Identification module **(120)** then scans the Omim Web pages for UID numbers, creating lists of UIDs that are associated with the microarray spots that had been linked to the Omim numbers. These steps are now described in more detail.

Specifically, for each Omim number stored in the LocusLink Data section **(144)** of the Data Repository **(138)**, the UID identification module **(120)** constructs a Web URL address of the form http://www.ncbi.nlm.nih.gov/htbin-post/Omim/dispmim?"Omimnumber", where "Omimnumber" is one of the Omim numbers mentioned above. The UID identification module **(120)** then requests this Web page by placing the Web address onto the Internet over the Internet Connection **(108)** and waits for a reply. It stores the reply on the disk **(106)**, saving it as hypertext markup language text with the name "Omimnumber.html".

After downloading all of the Omim Web page files corresponding to the Omim numbers that had been associated with spots on the microarray, the UID Identification module **(118)** extracts literature Unique IDentifier ("UID") numbers from these Web page files. It does so by scanning lines in each of the files for text of the form "db=m&form=6&dopt=d&uid=n1, n2, ...", where the integers n1, n2, ... are a sequence of UID numbers separated by commas. The "db=m" indicates that the UID numbers that follow are from the MedLine literature database. Then, the UID Identification module **(118)** extracts from the strings the numbers n1, n2, ..., which are separated by commas, which are bounded on the left by "&uid=", and which are bounded on the right by a character other than a comma or numerals 0 to 9. After all such UID numbers have been extracted from the Omim Web page, any repeated UID numbers are eliminated. In this preferred embodiment, this is done by sorting the numbers in ascending order using a standard sorting algorithm [Press et al (1992)], then examining them sequentially, removing an entry if it has the same value as the one that was examined previously. The sorted, non-duplicate UID numbers are then stored

by the UID Identification Module **(120)**, along with their corresponding Omim number in the section Literature UID Lists **(148)** of the Data Repository **(138)**.

Upon completion of these steps by the UID Identification Module **(120)**, the Process Control Module **(116)** initiates operation of the Text Acquisition Module **(122)**. Its function is to download over the Internet and filter the text files that are to be used to characterize the microarray data. When acquiring and filtering these data, the Text Acquisition Module **(122)** stores text data temporarily in the section Text Corresponding to UIDs **(150)** in the Data Repository **(138)**. For each UID corresponding to each Omim number, obtained from the section Literature UID Lists **(148)** of the Data Repository **(138)**, the UID Identification Module **(120)** constructs a Web URL address of the form http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?db=m&form=6&dopt=l&html=no&uid=UID,
where "UID" is one of the UID numbers. This Web URL address represents a query to obtain the literature citation, abstract, and indexing terms corresponding to the indicated UID number. The requested output format is MEDLARS, with hypertext formatting removed. Since March 2000, the base URL may also be http://www.ncbi.nlm.nih.gov/entrez/utils/qmap.cgi. (See Web address http://www.ncbi.nlm.nih.gov/entrez/utils/qmap_help.html for alternate ways to construct the URL). The Text Acquisition Module **(122)** then requests this Web page by placing the Web URL address onto the Internet over the Internet Connection **(108)** and waits for a reply. It stores the reply on the disk **(106)**, appending it to a text file having the name of the Omim number with which the UID is associated ("Omimnumber.txt"), with a delimiter line (like ">>UID Uidnumber") written to

separate successive UID downloads for that Omim number. All such files are stored by default in a specific directory (\Omim).

When all of the text corresponding to all of the UID numbers for all of the Omim numbers has already been downloaded, the User may also specify as an option that the system use those files for analysis, rather than download them all again.The Text Acquisition Module **(122)** may then also filter the text files, depending on the options that had been specified by the User at the User Interface **(104)**. The options are to filter out specific categories of text that are contained in the MEDLARS formatted Web pages. The categories of text are delimited in the files by labels such as AB - (for abstract), TI - (for the title of the literature citation), and MH - (for an index term characterizing the subject of the article), any or all of which can be retained as an option. For example, for the option of analyzing only the literature abstracts, the Text Acquisition Module **(122)** will scan the text files until the delimiter "AB -" is found and copy the subsequent text to another file containing the filtered text, until another delimiter is found. It then stops copying to the second file until another "AB-" is encountered. The latter file has the same name as the original file, plus an extension like ".AB" appended to the file name, indicating that it contains only the "abstract" category of text. Delimiter lines associating the filtered text with their UID numbers may also be retained as an option.

Clustering of Microarray Data

Upon completion of these steps by the Text Acquisition Module **(122)**, the Process Control Module **(116)** initiates operation of the Clustering Module **(124)**. Its

function is to organize the accession numbers (and therefore the associated UniGene numbers, LocusLink numbers, Omim numbers, and UID numbers and associated text files) into separate groups, known as clusters. The clusters are numbered 1, 2, 3, ... up to some maximum cluster number (Cmax).

If the clustering has already been performed external to the Computer Program Modules **(114)**, the User must have specified at the User Interface **(104)** the option of using externally clustered data. In that case, the Clustering Module **(124)** reads the name of the file that contains the clustering data (specified in the option by the User), then reads that file and places the data it contains in the section Clusters of Accession Numbers **(154)** in the Data Repository **(138)**. That data consists of the number of clusters (Cmax), as well as the accession numbers contained in each of the Cmax clusters. In principle, the same accession number could be found in more than one cluster, but most clustering algorithms place each accession number in only one cluster. In the event that some of the accession numbers in the section Accession Numbers **(140)** of the Data Repository **(138)** are not present in any of the Cmax clusters, they are placed in a cluster of previously ungrouped accession numbers, which is defined to be cluster number "zero".

If the microarray data have not already been clustered externally, the Clustering Module **(124)** performs the clustering itself. The clustering is performed using microarray data that must be provided in a file that had been specified by the User at the User Interface **(104)**. The microarray data are in the form of a spreadsheet table, in which each row corresponds to the respective accession numbers in the Accession Number section **(140)** of the Data Repository **(138)**, and the columns correspond to

different experimental conditions (e.g., response to different interventions, each measured at a single time point ; OR response to a single intervention at an initial time, measured at different subsequent times). The entry in each cell of the table is an integrated spot intensity (normalized so that reference spots have the same values for all conditions, and with unwanted background subtracted), measured in fluorescence or radioactivity units, for the microarray spot associated with that row's accession number. Alternatively, the entry in each cell of the table may be a dimensionless ratio, in which the numerator is the integrated spot intensity for that particular condition or time point, and the denominator is the baseline integrated spot intensity (value in the absence of the intervention, or value at time zero). It is also assumed that any other statistical corrections, such as those described by NEWTON et al., supra, have already been applied, so that any errors introduced by the measurement process itself are largely eliminated. After the microarray data are read from the input file, they are stored in the Microarray Data section **(152)** of the Data Repository **(138)**.

As an option, the User at the User interface **(104)** may instruct the Clustering Module **(124)** to add different amounts of noise to the microarray data described in the previous paragraph, in order to evaluate the robustness of the results that are eventually obtained without the addition of noise. The parameter of the option is the coefficient of variation of the statistical distribution that is sampled and added to the integrated spot intensity or ratio corresponding to each spot of the microarray. The statistical distribution is normal distribution with a mean equal to the given spot intensity and a standard deviation given by the mean times the coefficient of variation. Sampling of the statistical distribution makes use of standard random number

generator methods [PRESS et al., Numerical Recipes in C, Cambridge Univ. Press (1992)]. Because the spot intensity is constrained to be a non-negative number, in the event that the sampled random number is negative, it is then set equal to zero.

Clustering of these data by the Clustering Module **(124)** is performed using the algorithms described in the section "Background -- Description of Prior Art." The default method is an adaptation of the k-means algorithm, as implemented in the computer program CLARA [KAUFMAN et al., supra], source code for which was downloaded from the Web site http://lib.stat.cmu.edu/general/clusfind. When it was included in our system, CLARA was unchanged, except that we made it possible for CLARA to obtain parameter values for its algorithm from the Computer Program Modules **(114)**, rather than by independently prompting the User. Thus, when using this default algorthm for clustering, the User may specify the parameters for the CLARA program as options at the system's User Interface **(104)**, and the Clustering Module **(124)** subsequently runs the program CLARA using a system("...") function, as described earlier. In particular, the User may select the number of clusters that are to be constructed (Cmax). After the clustering has been performed, the Clustering Module **(124)** stores results in the the section Clusters of Accession Numbers **(154)** of the Data Repository **(138)**, namely, the number of clusters (Cmax), as well as the accession numbers contained in each of the Cmax clusters.

A special option is also available for the clustering of microarray data that are in the form of a time series, corresponding to a series of time points following the application of an intervention at the initial time point. Microarray data of this type are found, for example, at the Web site *genome-www.stanford.edu* and are described in

IYER et al., supra. They are from an experiment in which quiescent human diploid fibroblasts were stimulated to proliferate by increasing the concentration of fetal bovine serum in their growth medium at the initial time point. At 12 time points after addition of the serum, samples of mRNA were collected and used to prepare cDNA for hybridizing to an array. The purpose of the option for such time-series data is to perform the clustering based on estimated rates of mRNA transcription, rather than on total mRNA levels at the series of time points. Thus, as described now, the system performs signal processing to separate the contributions of mRNA synthesis and mRNA degradation from the measured actual mRNA levels, then performs the clustering with the estimated mRNA synthesis time series.

Let the letter i index each of the genes (spots) in the array, and let $x_i(t)$ denote the amount of mRNA corresponding to gene i at time t, expressed as a ratio relative to the value of $x_i(0)$, and corrected for any measurement errors such as unwanted background. The value of $x_i(t)$ at a sequence of time points will have been provided in a file specified by the User and is stored in the Microarray Data section **(152)** of the Data Repository **(138)**. The rate of change of x_i will be equal to its synthesis rate $f_i(t)$ minus its degradation rate $k_i(t)\, x_i$, where $k_i(t)$ is the degradation rate per mole of $x_i(t)$:

$$\frac{dx_i}{dt} = f_i(t) - k_i(t)\, x_i \qquad (1)$$

Our objective is to estimate $f_i(t)$, which will be used later instead of (or in addition to) $x_i(t)$ for purposes of clustering. To do so, first note that Eqn. 1 admits an integrating factor $I(t)$:

$$I_i(t) = exp \int_0^t k_i(\tau) d\tau \qquad (2)$$

The solution to Eqn. 1 is therefore equal to:

$$x_i(t) = \frac{x_i(0)}{I_i(t)} + \frac{1}{I_i(t)} \int_0^t f_i(\tau) I_i(\tau) d\tau \qquad (3)$$

To perform the integration in Eqn. 3, we approximate $f_i(t)$ piecewise as lines between time points at which the data were sampled, as a first approximation truncating its Taylor series at the linear term:

$$f_i(t) = f_i(T_j) + \sigma_{ij} [t - T_j] + \ldots \quad for \quad T_j \leq t < T_{j+1} \qquad (4)$$

To perform the integration in Eqn. 3, we must also model the degradation function $k_i(t)$, then substitute that function into Eqn. 2. When $k_i(t)$ is approximated piecewise as lines between time points at which the data were sampled, as in Eqn. 4, then the truncated Taylor series for $k_i(t)$ is

$$k_i(t) = k_i(T_j) + \omega_{ij} [t - T_j] + \ldots \quad for \quad T_j \leq t < T_{j+1} \qquad (5)$$

and Eqn. 3 becomes

$$x_i(t) = x_i(T_j) e^{-\{k_i(T_j)\ [t-T_j] + \frac{w_{ij}}{2}[t-T_j]^2\}} + f_i(T_j) \int_0^{t-T_j} d\tau\ e^{-\{k_i(T_j)\ ([t-T_j]-\tau) + \frac{w_{ij}}{2}([t-T_j]^2 - \tau^2)\}}$$

$$+ \sigma_{ij} \int_0^{t-T_j} d\tau\ \tau e^{-\{k_i(T_j)\ ([t-T_j]-\tau) + \frac{w_{ij}}{2}([t-T_j]^2 - \tau^2)\}} \quad for \quad T_j \leq t < T_{j+1} \qquad (6)$$

Successive segments of the synthesis rate function $f_i(t)$ in Eqn 4 are joined together to make it a continuous function by letting $f_i(0)$ of a segment equal $f_i(T_{max})$ of the previous time segment. All parameters of the synthesis function may be estimated from the data, given $k_i(T_j)$, with the exception of $f_i(0)$ for the first segment, which can be set equal to a value representing an initial steady state, $k_i(0)\ x_i(0)$. Thus, given the experimental data $x_i(T_j)$ and the values of $k_i(T_j)$ (and therefore the values of ω_{ij} =

$[k_i(T_{j+1}) - k_i(T_j)]/[T_{j+1} - T_j]$), Eqn. 6 may be solved numerically for σ_{ij} (and therefore the values of $f_i(T_{j+1}) = f_i(T_j) + \sigma_{ij} [T_{j+1} - T_j]$) over successive intervals between time points at which the data were sampled. This is done by setting $t = T_{j+1}$ in Eqn. 6, rearranging Eqn. 6 to obtain an expression for σ_{ij}, then evaluating the resulting expression by integrating numerically, using Romberg's method [PRESS et al., supra].

Similarly, successive segments of the degradation function in Eqn. 5 are joined together to make it a continuous function by letting $k_i(0)$ of a segment equal $k_i(T)$ of the previous segment. All parameters of the degradation function may be estimated from the microarray data, given $f_i(T_j)$, with the exception of $k_i(0)$ for the first segment, which can be set equal to a value representing an initial steady state, $f_i(0)/x_i(0)$. Thus, given the experimental data $x_i(T_j)$ and the values of $f_i(T_j)$ (and therefore the values of $\sigma_{ij} = [f_i(T_{j+1}) - f_i(T_j)]/[T_{j+1} - T_j]$), Eqn. 6 may be solved numerically for ω_{ij} (and therefore the values of $k_i(T_{j+1}) = k_i(T_j) + \omega_{ij} [T_{j+1} - T_j]$) over successive intervals. We use Romberg's method for the integrals and Brent's method for finding the value of ω_{ij} that satisfies Eqn. 6 [PRESS et al., supra]. Note that f and k are defined to be non-negative, so if the calculations were to yield a negative value, the value is set equal to zero.

Thus, if the degradation function (Eqn. 5) is known for a particular gene, we can calculate its synthesis function (Eqn. 4), and if the synthesis function is known, we can calculate the degradation function.

Parameters in the model described above may be estimated using the EM or related algorithms, and it is useful to initialize such algorithms by first obtaining results from a preliminary calculation [MANNING et al., Chapter 14, "Clustering", in Foundations of Statistical Natural Language Processing, MIT Press (1999)]. Our method for doing so is as follows. Initially, we do not know what the degradation functions are for any of the genes represented in the microarray, so our method is to first make a large number of guesses for the degradation functions, calculate the corresponding synthesis functions using Eqn. 6 or its equivalent, then cluster the synthesis functions.

Several degradation models are now described, all of which are used for the initial guess functions. The simplest model assumes that k_i is a constant, such that the half-life of the mRNA for gene *i* ($\ln 2 / k_i$) is determined primarily by the gene's sequence. In eukaryotic cells, mRNA that is richer in AU nucleotides in its 3' untranslated region has a half-life that is shorter than mRNA species that are poorer in AU nucleotides. When the simplest model is used, k_i may then be taken outside of the integral in Eqn. 2, and *I(t)* becomes a simple exponential. Eqn. 6 may then be evaluated analytically to give $x_i(t)$ over a time segment of duration *T*, the beginning and end of which correspond to data points:

$$x_i(t) = x_i(T_j) + \frac{\sigma_{ij}}{k_i}[t - T_j] - [x_i(T_j) - \frac{f_i(T_j)}{k_i} + \frac{\sigma_{ij}}{k_i^2}][1 - e^{-k_i[t-T_j]}] \quad for\ T_j \leq t < T_{j+1} \quad (7)$$

After rearranging this equation, the value of the synthesis parameter σ for each segment may be estimated from the data as follows, eliminating the need for numerical root solving of Eqn. 6:

$$\sigma_{ij} = \frac{k_i^2[x_i(T_{j+1}) - x_i(T_j)] + k_i[k_i x_i(T_j) - f_i(T_j)][1 - e^{-k_i[T_{j+1}-T_j]}]}{k_i[T_{j+1} - T_j] - [1 - e^{-k_i[T_{j+1}-T_j]}]} \quad for\ T_j \leq t < T_{j+1} \tag{8}$$

A globally modulated model for $k_i(t)$ acknowledges that although mRNA species degrade at rates that are a function of their individual sequences, degradation also depends on factors such as the concentrations of ribonucleases that may change in time. Similarly, the activity of enzymes responsible for cytosolic polyadenylation of mRNA may change in time, thereby modulating the stability of mRNA in the cytoplasm. The analysis of this model is like the the simplest model, except that the values of the $k_i(t)$ are all modulated by a function $f(t)$ -- $k_i(t)=\kappa_i [1 + f(t)]$, where κ_i is constant for gene *i* as in the simplest model.

An mRNA-specific modulated model for $k_i(t)$ also acknowledges that mRNA degradation may depend on factors that may change in time, but unlike the globally modulated model, it permits the modulation to be specific for individual mRNA species. A model of this type is $k_i(t)=k_i [1 + a\ f_i(t)]$ where k_i is constant for gene *i* as in the simplest model, and where the parameter *a* may be positive or negative. For example, steroid hormones such as testosterone not only increase the transcription of specific genes, they also increase the stability of many of the mRNAs encoded by those genes. This model, with a negative value for *a*, would represent that situation. The model may also be combined with the previous model to allow some of the modulation to be global and some to be specific for the particular mRNA species.

Synthesis rates are calculated for each of these initial guess models, with multiple parameter values for each of them, such as different values of k_i for the simplest model (default number of parameter values = 100, which may be changed as an option). The range of values for the parameters to be tried (or functional values in the case of the trend model) is also an option that may be specified by the User. The synthesis rates resulting from all of the initial guess models for all of the genes on the microarray are then clustered. Wherever different genes agree about a synthesis function, as evidenced by the formation of a tight cluster (formed by a standard clustering algorithm, such as CLARA described above), the corresponding degradation functions are tentatively accepted. Otherwise, the initial guesses for degradation functions are discarded on the grounds that the synthesis function for a particular gene and parameter value is farther from the centroid than the other synthesis functions corresponding to different initial guess parameter values for that same gene. These degradation functions are discarded in steps, with reclustering performed before the next step of discarding. When only one of the initial guess degradation functions remain for each of the genes, this clustering process is complete. The synthesis function corresponding to the remaining degradation function (and particular parameter value) will belong to a synthesis function cluster. The synthesis function that is estimated for each gene in the cluster is the centroid of the cluster.

As an option, clustering my also be allowed to continue in an iterative fashion, as now described. The synthesis functions for different genes that eventually remain in a cluster may be used as multiple initial guesses for estimating the degradation functions

for all of the genes in that cluster, using Eqn. 6 or its equivalent. The resulting degradation functions may in turn be used to estimate new synthesis functions, which again are clustered, and guesses for the degradation function are again discarded on the grounds that they produce synthesis functions that are too far from the centroid of a cluster. Thus, the process of alternately calculating synthesis and degradation functions continues iteratively until the genes' membership in clusters does not change. At that point, the synthesis function for each gene in a cluster is taken to be its centroid, with an uncertainty determined by the range of functions in the cluster.

Text Modeling and Classification

Upon completion of these steps by the Clustering Module **(124)**, the Process Control Module **(116)** initiates operation of the Text Modeling Module **(126)**. Its first function is to group the text associated with accession numbers -- the text having been obtained with the Text Acquisition module **(122)** -- so as to correspond to the clusters in the Data Repository section Clusters of Accession Numbers **(154)**. It then produces a statistical model of the text that is suitable for text classification. The model contains, for example, the number of times that a term like "mitosis" appears in different documents. To do so, it first creates a directory called \clusters_text, and within this directory it creates subdirectories \0, \1, \2, ... , \Cmax corresponding to each of the clusters. For each of the accession numbers in data section Accession Numbers **(140)**, it then copies files produced by the Text Acquisition Module **(122)** into the subdirectories, according to the cluster to which that accession number has been assigned by the Clustering Module **(124)**. If an accession number could not be

associated with an Omim number, or if that Omim number could not be associated with any UIDs, then the Text Modeling Module **(126)** proceeds to the next accession number without copying files. The Text Modeling Module **(126)** then performs the following system function (written in the C programming language):

system("rainbow -d .\\text_model --index .\\clusters_text*");

to instruct the computer program 'rainbow' to use the text found in subdirectories of the directory \clusters_text, make a statistical model of the text, then store the results in a directory called \text_model [McCALLUM, supra]. Results are also stored in the Data Repository **(138)** in the section Statistical Model of Text **(156)**. When making the statistical model, the rainbow program turns the stream of characters in each file into words called "tokens". By default, all sequences of characters are converted to lowercase (e.g., "A" becomes "a") to form the tokens, and any token that is in a stoplist of common words (e.g., "the", "of", "is") are ignored. The model that is then formed constitutes a sparse matrix of the number of times that each of the tokens is found in each of the text files in each of the subdirectories. Many options for the rainbow computer program may also have been specified by the User at the User Interface **(104)**, which are subsequently passed to the rainbow computer program by the Text Modeling Module (126) using a system("...") function. The possible options are listed in the documentation for the 'rainbow' computer program at the Web site *www.cs.cmu.edu/~mccallum/bow*, as well as its "rainbow --help" on-line documentation, which are included here by reference. Three examples of those options are as follows. (1) Pass the words of the files through a stemmer (e.g., change "experimenting" to "experiment"). (2) Create N-gram tokens (e.g., with a bi-gram,

"cell cycle" is used as a token, not just individual words, "cell" and "cycle"). (3) Prune the word list by ignoring words that occur less than some specified number of times in the text files.

Once the text has been modeled by the Text Modeling Module **(126)**, the Process Control Module **(116)** initiates operation of the Keyword Identification Module **(128)**, which uses the rainbow computer program to acquire various diagnostic information about the model. The information that is provided automatically by the Keyword Identification Module **(128)** is a list of words for each cluster, sorted in descending order according to the numerical weights calculated by a classification algorithm. The following system function, written in the C programming language, is used by the Keyword Identification Module **(128)** to generate the word lists:

system("rainbow -d .\\text_model --print-word-weights=cluster_name>word-weights_cluster_name.txt");

where cluster_name is the name of a cluster (0,1, 2,..., or Cmax) and where word-weights_cluster_name.txt is the name of a text file that is to contain the words and corresponding weights for that particular cluster. Word lists for each of the clusters are generated in succession by performing a system command of this form, but with different cluster names (i.e., cluster numbers). The results are stored on the storage disk **(106)** in the indicated word-weight file, the contents of which are read into the section Key Words or Phrases **(158)** in the Data Repository **(138)** by the Keyword Identification Module **(128)**. By default, only words with positive weights are read into the Key Words or Phrases section **(158)**, sorted in descending order according to

the weight values. As an option, words with negative word weights may be included as well. Examples of such word lists are given in Tables 1, 2, 3, and 4.

Additional word-list diagnostic information may be requested as an option by the User at the User Interface **(104)**. Those options [McCALLUM, supra] are available through the rainbow computer program, for example, a list of words that have the highest log odds ratio score for each cluster. The default number of words is 20 per cluster, obtained by the following system function in the C programming language: system("rainbow -d .\\text_model --print-log-odds-ratio=20>log_odds_word_list.txt"); The results are stored on the storage disk **(106)** in the file log_odds_word_list.txt, the contents of which are read into the section Key Words or Phrases **(158)** in the Data Repository **(138)** by the Keyword Identification Module **(128)**. Another example of a word list option is a list of words having the highest mutual information with the cluster index, sorted by the magnitude of the mutual information. A description of such options is present in the on-line documentation for the rainbow program, which is included here by reference.

Upon completion of these steps by the Keyword Identification Module **(128)**, the Process Control Module **(116)** initiates operation of the Text Classification Module **(130)**. As described in a previous paragraph, files produced by the Text Acquisition Module **(122)** were copied into subdirectories \0, \1, ..., \Cmax, depending on the cluster to which the corresponding accession numbers had been assigned by the Clustering Module **(124)**. The Text Classification Module **(130)** divides the text files in each of these directories into two sets -- a training set and a testing set. The Text Classification Module **(130)** then learns to predict the cluster number of a text file

from the words that it contains, based on the statistical distribution of words present in the training set of files, or more precisely, tokens from those files that are present in the Statistical Model of Text **(156)** that had been generated by the Text Modeling Module **(126)**. That is to say, the Text Classification Module **(130)** performs supervised machine learning because it knows the cluster number that is associated with each of the training set files (i.e., the subdirectory in which each of the training set files is found). It then tests its ability to predict the cluster number, using the testing set of files as input. For purposes of predicting the cluster number of a test file, the Text Classification Module **(130)** uses only information from text in that file. It then compares the results of its prediction with the actual cluster number associated with that file. After performing the prediction for all of the files in the testing set, it constructs a confusion matrix. Rows of that matrix (indexed with i) refer to the actual cluster number associated with the test file. Columns of the matrix (indexed with j) refer to the cluster number that was predicted. Elements of the matrix are the number of times that cluster number j was predicted by the classifier, given that the actual cluster number was i. Perfect classification would occur only if the matrix consisted of zeros except along the diagonal.

For detailed analysis of the classification of particular files in the test sets, the number of times that each particular file was correctly classified, and the number of times that it was incorrectly classified, is also recorded. The files with the highest percentage of correct classifications, as determined by sorting the percentages for all files, ranks those files as being the most relevant to the subject matter that is characteristic of the cluster. Consequently, the accession numbers with which those

highly ranked files are associated are also ranked as being the most relevant to the subject matter that is characteristic of the cluster.

The Text Classification Module **(130)** implements these steps using the rainbow program. It does so using a system function in the C programming language of the form:

system("rainbow -d .\\text_model --test-set=0.4 --test=100 | perl confusion_matrix > confusion_matrix.txt");

where the rainbow option -d .\\text_model indicates the directory location of the text statistics file that was created by the Text Modeling Module **(126)**.

where the rainbow option --test-set=0.4 indicates that 40% of the text files in the subdirectories \0, \1, ... \Cmax are to be used for testing and the remaining 60% are to be used for training the classifier. The files in the test/train split are selected at random.

where the rainbow option --test=100 indicates that the process of selecting the test/train files, followed by classification, is to be performed 100 times.

and where the results are passed to the perl script "confusion_matrix", which writes each of the 100 confusion matrices to a file named confusion_matrix.txt. This script is the same as the perl script "rainbow_stats" in the rainbow source distribution at *www.cs.cmu.edu/~mccallum/bow*, which is hereby incorporated by reference, except that the script "confusion_matrix" does not write as output any information other than the confusion matrix (lines for writing out the other information are deleted from rainbow_stats to make confusion_matrix). The confusion matrix data are also stored in the Text Classification Data section **(160)** in the Data Repository **(138)**.

The default method that the program rainbow uses for classification is the Naive

Bayes method, otherwise known as Evidence Classification or simply the Bayes method. This method, along with applications related to text classification, is explained in Chapter 6 of MITCHELL, Machine Learning, McGraw-Hill (1997). The User at the User Interface **(106)** may also specify as an option that a different method be used by the computer program 'rainbow' to perform the classification, including support vector machines (svm), term frequency- inverse document frequency (tfidf), probabilistic indexing (prind), maximum entropy (maxent), k-nearest neighbors (knn), EM algorithm (em), Dirichilet kernel (dirk), and Active Learning (active). These options are then requested by the Text Modeling Module **(126)** by issuing a

system("...--method=METHOD...")

command for the indicated options to be performed by the rainbow computer program, where METHOD is one of the methods indicated above in parentheses.

Upon completion of these steps by the Text Classification Module **(130)**, the Process Control Module **(116)** initiates operation of the Cluster Randomization Module **(132)**. The function of this module is to repeat the steps performed by the Text Modeling Module **(126)** and Text Classification Module **(130)**, except that the accession numbers associated with the clusters are randomized. The objective of this randomization is to ascertain the extent to which the results that had been obtained with the Text Classification Module **(130)** could have occurred by chance. The number of clusters (Cmax) and the number of accession numbers within each cluster is not changed by the Cluster Randomization Module **(132)**. Instead, this module takes the list of Accession Numbers in the Data Repository **(138)** and shuffles their order randomly using a random number generator [PRESS et al., supra]. It then replaces the

existing accession numbers associated with the clusters, using the shuffled list of accession numbers. The Process Control Module **(116)** then causes the previous steps performed by the Text Modeling Module **(126)** and by the Text Classification Module **(130)** to be repeated. The confusion matrix ouput for the randomized cluster data are then stored in the Text Classification Data section **(160)** of the Data Repository **(138)**. The process of randomizing the accession numbers, text modeling, and text classification is then repeated many times (default number =100), the results of which are also stored in the Text Classification Data section **(160)** of the Data Repository **(138)**. The User at the User Interface **(106)** may also select the number of randomizations as an option, rather than use the default number.

Upon completion of these steps, the Process Control Module **(116)** initiates operation of the Data Summarization Module **(134)**. The purpose of this module is to extract a summary of the confusion matrix data present in the Text Classification Data section **(160)** of the Data Repository **(138)**. Those matrix data give the number of times that each of the test files were correctly classified or were incorrectly classified as another cluster number. The Data Summarization Module **(134)** first divides the number of correctly classified files for each cluster by the total number classified, giving the fraction of correctly classified files for each of the many (default=100) randomly sampled test/train splits of the originally clustered data, as well as for each of the trials in which the clusters' accession numbers were randomized. It then sorts those values for the original clustering in ascending order, to provide an estimated a statistical distribution for the fraction of correctly classified files, for each of the clusters. It also sorts in ascending order the values obtained from the trials in which

the accession numbers were randomized, to provide an estimated control statistical distribution for the fraction of correctly classified files for each cluster. A Kolmogorov-Smirnov statistical test is then applied to these two sets of data for each cluster, to test the null hypothesis that the two data sets are drawn from the same distribution [PRESS et al., supra].

For each cluster, the mean of the statistical distribution for the fraction of correctly classified files is calculated. This is done for the distribution corresponding to the originally assigned accession numbers, as well as for the distribution corresponding to the randomized accession numbers. The difference between the means (original minus randomized) is then calculated as an overall figure-of-merit index for the quality of the clustering.

Similarly, for each cluster i, the mean of the statistical distribution (original minus randomized accession numbers) for the fraction of files incorrectly classified as cluster j is calculated, where i and j range from 1 to Cmax, and where i is not equal to j. These values indicate the closeness of functional relationship between each pair of clusters, as indicated by the literature about the clusters. Thus, if text about cluster i is frequently misclassified as text about cluster j, this indicates that the genes in clusters i and j have functions in common.

Upon completion of these steps by the Data Summarization Module **(134)**, the Process Control Module **(116)** initiates operation of the Data Output Module **(136)**. It displays the key words or phrases, and it also provides the results of the figure of merit calculations and the Kolmogorov-Smirnov statistical tests for each of the clusters. Finally, it provides a graphical display of the text classification results, which gives an

indication of the extent to which, on average, the text about each gene within a cluster resembles that of other genes within a cluster, as compared with that of genes selected at random from all the clusters. Examples of the format of that graphical display are given in **Fig. 3** and **Fig. 4**.

DESCRIPTION OF ALTERNATE EMBODIMENTS

The alternate embodiment of the system for analyzing microarray data, shown in **Fig. 2**, is the same as the preferred embodiment (shown in **Fig. 1**) except that:

(i) the following computer program modules in **Fig. 1** are not used -- Text Acquisition Module **(122)**, Text Modeling Module **(126)**, Keyword Identification Module **(128)** and Text Classification Module **(130)**;

(ii) the following sections of the Data Repositiory in **Fig. 1** are not used -- Text Corresponding to UIDs **(150)**, Statistical Model of Text **(156)**, Key Words and Phrases **(158)** and Text Classification Data **(160)**;

(iii) A program module called the Citation Analysis Module **(200)** and a section of the Data Repository called Coupling Strength Data **(202)** are found in the alternate embodiment **(Fig. 2)**, but not in the preferred embodiment **(Fig. 1)**.

Operation of the alternate embodiment is the same as the preferred embodiment during the initial steps of the process for analyzing microarray data. The accession numbers are converted to UniGene numbers, which are converted to one or more

LocusLink numbers, which are converted to one or more Omim numbers. The system then automatically downloads over the Internet the Web pages associated with the Omim numbers and extracts from them PubMed unique identifiers (uids) associated with those Omim numbers, i.e., a list of publications that can be closely associated with each spot in the microarray. It then removes duplicate uids associated with individual Omim numbers.

For the alternative embodiment shown in **Fig. 2**, the Process Control Module **(116)** skips the Text Acquisition Module **(122)** in **Fig. 1** and proceeds to perform the steps of the preferred embodiment for the Clustering Module **(124)**. As with the preferred embodiment, after the clustering has been performed, the Clustering Module **(124)** stores results in the the section Clusters of Accession Numbers **(154)** of the Data Repository **(138)**, namely, the number of clusters (Cmax), as well as the accession numbers contained in each of the Cmax clusters.

For the alternate embodiment, the Process Control Module **(116)** then initiates operation of the Citation Analysis Module **(200)** in **Fig. 2**. The function of the Citation Analysis Module **(200)** is to perform a type of 'bibliographic coupling' analysis [EGGHE and ROUSSEAU, Introduction to Informetrics, Elsevier Science Publ. (1990)], in which the coupling strength between 'articles' (Omim Web pages for different genes, where each gene is represented by one Omim Web page) is a function of how many times they cite the same 'paper' (PubMed uid). The method for doing so is now described.

First, the Citation Analysis Module **(200)** organizes data by creating a directory on the Storage Disk **(106)** called 'cluster_uids' and places in it Cmax subdirectories corresponding to each of the clusters: \0, \1, ...\Cmax. In each of the subdirectories it creates a file having the name of each Omim number associated with each of the accession numbers that are associated with the corresponding cluster. The contents of each such file are a list of the uid numbers associated with the corresponding Omim number.

To do so, the Citation Analysis Module **(200)** combines the following data already present in the Data Repository **(138)**. The Clustering Module **(124)** had stored the accession numbers corresponding to each cluster in the the section Clusters of Accession Numbers **(154)** of the Data Repository **(138).** The Initialization Module **(118)** had linked the Accession Numbers in **(140)** to Omim numbers by following the pointers to UniGene numbers, then to LocusLink numbers, then to Omim numbers. Furthermore, the sorted, non-duplicate UID numbers were stored by the UID Identification Module **(120)**, along with their corresponding Omim number in the section Literature UID Lists **(148)** of the Data Repository **(138)**.

Then, for every possible pair of Omim numbers within each of the subdirectories \0, \1, ...\Cmax, the Citation Analysis Module **(200)** goes down the list of uid numbers associated with one member of the pair. As it goes, it checks to see whether that uid number is also found in the list of uid numbers associated with the other member of the Omim pair. The total number of matches that it finds is then divided by the total number of uid numbers associated with one or the other Omim number, whichever of

the two have the shorter list. This ratio is defined as the 'coupling strength' between the two Omim numbers.

These coupling strength values, which are associated with each possible pair of Omim numbers (within a subdirectory \0, \1, ..., or \Cmax) are then stored in the section Coupling Strength Data **(202)** in the Data Repository **(138)**. When all the coupling strengths are calculated for the pairs of Omin numbers in a subdirectory corresponding to a cluster, these coupling strengths are then averaged to provide an average coupling strength for that cluster, which is also stored in the section Coupling Strength Data **(202)** in the Data Repository **(138)**.

When all the average coupling strength values have been calculated, the Process Control Module **(116)** passes control to the Cluster Randomization Module **(132)**. It randomizes the assignment of accession numbers to clusters, using the same method that was described in connection with the preferred embodiment. The Process Control Module **(116)** then passes control back to the Citation Analysis Module **(200)**. The process of calculating an average coupling strength for each of the clusters is then repeated as described above, namely, the association of Omim numbers to the clusters based on their (randomized) accession number composition, the estimation of coupling strength for each pair of Omim numbers in each cluster, and the calculation of an average coupling strength for each cluster. Those results are also stored in the Coupling Strength Data section **(202)** of the Data Repository **(138)** for subsequent use. The entire process of accession number randomization and coupling strength calculation is repeated the number of times that was specificed as an option by the User (default number of times=100).

The Process Control Module **(116)** then passes control to the Data Summarization Module **(134)**. For each cluster, the average coupling strengths for the randomized clusters are sorted in ascending order. Then, as an index of the sigificance of the average coupling strength that had been obtained before randomizing the clusters, the Data Summarization Module **(134)** determines the percentage of times that the observed coupling strength for each cluster was larger than those obtained with the randomized clusters. For example, if exactly half of the average coupling scores obtained by randomizing the cluster was less than the average coupling score before randomization, the percentage coupling score for that cluster would be 50%. Those percent coupling scores are then stored in the Summarized Data section **(162)** of the Data Repository **(138)** for subsequent display by the Data Output Module **(136)**.

EXAMPLES USING THE PREFERRED EMBODIMENTS

The examples make use of microarray data that are described in IYER et al.,, supra, which are available at the Web site *genome-www.stanford.edu*. They are from an experiment in which quiescent human diploid fibroblasts were stimulated to proliferate by increasing the concentration of fetal bovine serum in their growth medium. At 12 time points after addition of the serum, samples of mRNA were collected and used to prepare cDNA for hybridizing to an array. Five hundred-seventeen genes on the array were observed to show significant up- or down-regulation, as defined in IYER et al., supra. The microarray data corresponding to these 517 genes were clustered using a hierarchical clustering algorithm [EISEN et al., supra], resulting in 10 clusters, labeled A through J in fig. 2 of IYER et al., supra.

Accession numbers for all the genes represented as spots on the microarray are available at the Web site given above.

Analysis of these data made use of information extracted from Build 96 of the Unigene database, which was contained in the file Hs.data.Z, downloaded from the Web site *ftp.ncbi.nlm.nih.gov/pub/schuler/Unigene.* The analysis also made use of information extracted from the Locus Link database (version September 10, 1999), which was downloaded as the file LL_templ at the Web site *ftp.ncbi.nlm.nih.gov/refseq/LocusLink*. Processing of these data to associate the accession numbers with Omim numbers proceeded as described in the Preferred Embodiment, through pointers from accession numbers to UniGene numbers to LocusLink numbers to Omim numbers. Among the 517 accession numbers in the test data, 237 of them could be associated with Omim numbers. Most of the others were unidentified ESTs, which were excluded from further analysis by associating them with Omin number "0".

The UID Identification Module **(120)** in **Fig. 1** then downloaded Omim Web pages corresponding to those Omim numbers and extracted from them Unique IDentifier numbers ("uids"), as described in the Preferred Embodiment. A total of 19,643 uids were associated with 237 Omim numbers.

The Text Acquisition Module **(122)** in **Fig. 1** then downloaded text files corresponding to these 19,643 uids, storing them in 237 files having the names of the corresponding Omim numbers, as described in the Preferred Embodiment.

The Clustering Module **(124)** in **Fig. 1** then clustered the 517 accession numbers using an externally available clustering scheme, which is the one shown in Fig. 2 of

IYER et al., supra. The clusters, associated with the accession numbers of the clustered genes, were obtained at the Web site http://genome-www.stanford.edu. For use in the example, the clusters that IYER et al., supra, had labeled A through J were relabeled 1 through 10, and accession numbers that had not been associated with a cluster were put into a cluster "zero".

In a second analysis, the Clustering Module **(124)** in **Fig. 1** also clustered these 517 accession numbers, this time clustering by itself, by estimating mRNA synthesis rates for each of the microarray spots corresponding to these 517 accession numbers, then clustering those estimates with the CLARA clustering algorithm that is described in the Preferred Embodiment. To do so, the Clustering Module **(124)** used the actual microarray data as input to the clustering algorithm. The data for the clustering were downloaded from the Web site http://genome-www.stanford.edu. The gene expression changes found at that Web site are given as the ratio of the expression level at the given time-point to the expression level in serum-starved fibroblasts. All ratios are given as normalized to time zero, and the time points correspond to 0.25, .5, 1, 2, 4, 6, 8, 12, 16, 20 and 24 hours after a shift in the serum concentration. For clustering by the CLARA algorithm, options were that there would be 10 clusters. The result of the clustering was that most of the accession numbers were grouped into only 5 clusters, with the remaining 5 clusters containing less than 10 accession numbers each.

In a third analysis, the data in the second analysis were again used, but before processing these data, the Clustering Module **(124)** added noise to each of the values for each of the microarray spots, in order to investigate the effects of added noise on the results. The coefficient of variation for the the added noise was 30%.

The Text Modeling Module **(126)** processed the text associated with the accession numbers of the examples, as described in the Preferred Embodiment section, followed by processing by the Keyword Identification Module **(128)**. For the data clustered externally as given in IYER et al., supra, examples of the list of Keywords characterizing the clusters are shown in **Tables 1** and **2**. Table 1 gives keywords for Cluster D, and Table 2 gives keywords for Cluster B. For the data clustered by the Clustering Module **(124)** itself after estimating mRNA synthesis rates, examples of the list of Keywords characterizing the clusters are shown in **Tables 3** and **4**. Table 3 gives the keywords for a cluster when no noise was added to the microarray data before processing it. Table 4 gives the keywords for the comparable cluster, produced when noise was added to the microarray data before processing.

TABLE 1: TOP 25 KEYWORDS FOR CLUSTER D.

WEIGHTS	KEYWORDS
0.012058684	msh
0.007730326	hnpcc
0.004746247	mismatch
0.004467339	colorectal
0.004294871	arf
0.002764531	hmsh
0.002495748	crm
0.002355605	kinetochore
0.002203363	mlh

0.002048244 ebna

0.001990554 nonpolyposis

0.001818067 hmlh

0.001784532 export

0.001580882 nes

0.001388732 mmr

0.001341535 msi

0.001280268 topoisomerase

0.00124356 spindle

0.001199594 ebv

0.001149875 primase

0.001112527 repair

0.001073333 kinetochores

0.001050086 mad

0.001013135 replication

0.000948283 muts

TABLE 2: TOP 25 KEYWORDS FOR CLUSTER B.

WEIGHTS	KEYWORDS
0.006132082	caveolin
0.004281967	kit
0.002052745	dpd
0.00190056	mast

0.001568304 mastocytosis

0.00155767 hsf

0.001361383 icc

0.001161008 adducin

0.001126881 tropomodulin

0.001027287 caveolae

0.000914975 syntaxin

0.000914036 hox

0.000884391 btk

0.000863388 tfiib

0.000795634 fyn

0.00074206 wee

0.000691434 kindling

0.000666647 ciita

0.000600745 fucose

0.000548474 fu

0.000506843 chop

0.000476232 sls

0.000416166 meis

0.000395423 ahr

0.000365585 rab

TABLE 3. TOP 25 KEYWORDS FOR CLUSTER OBTAINED THROUGH ESTIMATION OF MRNA SYNTHESIS RATES

WEIGHTS	KEYWORDS
0.005203617	vegf
0.001728392	kgf
0.001462155	tenascin
0.001232543	fgf
0.001063043	elastin
0.001015313	hex
0.001010644	udp
0.000831199	hexosaminidase
0.000798481	dystonia
0.000784139	pdgf
0.000709865	plasminogen
0.000672346	cpe
0.000642022	pai
0.000563102	timp
0.000493248	abp
0.00045766	gro
0.000412615	glcnac
0.0004094	cadherin
0.000396513	pk
0.000358335	cyclohydrolase

0.000348477 gpib

0.000296712 prad

0.000260455 thrombomodulin

0.00025931 stc

0.000244859 infarction

TABLE 4. TOP 25 KEYWORDS FROM CLUSTER OBTAINED BY ADDING NOISE TO MICROARRAY DATA

WEIGHTS KEYWORDS

0.005203734 vegf

0.001661089 tenascin

0.001538653 elastin

0.001167705 hex

0.001150809 udp

0.000957946 hexosaminidase

0.000957585 fkbp

0.000825281 plasminogen

0.000809836 dystonia

0.000791792 pai

0.000655273 endothelin

0.000555229 timp

0.000553542 cadherin

0.000548997 gro

0.000485873 glcnac

0.000421891 cyclohydrolase

0.000366406 fk

0.00033571 thrombomodulin

0.000322609 abp

0.000311342 stc

0.00029394 infarction

0.000269864 arterial

0.000266684 galactosyltransferase

0.000260124 urokinase

0.000255695 arch

The Text Classification, Cluster Randomization, and Data Summarization Program Modules (**130**, **132**, and **134**, respectively) then processed the data as described in the Preferred Embodiment. The Data Output Module **(136)** then presented the results. **Fig. 3** and **Fig. 4** show representative results when the accession numbers were clustered, as given external to the system in IYER et al., supra. The clusters correspond to the ones whose Keywords were given in Tables 1 and 2, respecively.

Figure 3 shows graphical results for the fourth cluster (Cluster "D") in IYER et al., supra. As described in the Preferred Embodiments section, text about the genes corresponding to accession numbers were randomly split into testing and training sets ("test/train splits"). A Bayes model was used to model the words in the training set. The model was then used to predict the cluster corresponding to text in the documents

in the testing set. The fraction of test documents correctly classified (in Fig. 3, corresponding to Cluster D) was then calculated. The documents in the test/train split were randomized 100 times, so the fraction of correctly classified documents (in Fig. 3, corresponding to Cluster D) was also calculated 100 times. The fractions so obtained were sorted in ascending order, and their values are shown in Fig. 3 (Using Original Accession Numbers). The accession numbers associated with the clusters were also randomized, and the entire process involving test/train splits was repeated 100 times to generate the values also shown in Fig. 3 ("Using Randomized Accession Numbers"). The graph using the original accession numbers lies significantly above the graph using randomized accession numbers, indicating that the text files corresponding to the original accession numbers for Cluster D are much more predicitive of the word content of one another, than text files corresponding to random accession numbers. The means of these two distributions are calculated by the system and the difference between the two means (0.302) is taken to be a figure of merit for the quality of the cluster. The results shown in Fig. 3 are representative of most of the 10 clusters shown in IYER et al., supra, in the sense that the fraction of correctly classified test documents has a statistical distribution that is consistently larger for the original clustering than for accession numbers randomly assigned to the cluster. The probability that the two distributions in Fig. 3 are the same is less than 10^{-16}, according to the Kolmogorov-Smirnov test results. The conclusion that may be drawn from this figure is that text about different genes within the cluster uses the same words much more often than text about randomly selected genes, for example the words shown in **Table 1**. The User may therefore conclude from the results shown in **Fig. 3** that the

genes represented by this cluster are functionally related, as evidenced by the same words being used in literature about those genes. The words in **Table 1** suggest the nature of the concepts that genes in that cluster have in common, namely, genes, structures, and functions related to chromosome replication and repair.

Figure 4 shows graphical results for the another cluster (Cluster "B" in IYER et al., supra). These results are atypical of the clusters in the sense that the fraction of correctly classified test documents has a statistical distribution that is NOT consistently larger for the original clustering than for genes randomly assigned to the cluster. In fact, the difference between the means of the two distributions shown there is only -0.002, a figure of merit that is much worse than the ones for most of the other clusters.The general interpretation of these results is that on average, text about genes within this cluster uses the same words no more often than words found in text about the randomly-selected accession numbers. The User may therefore conclude from the results shown in **Fig. 4** that the clustering method used in IYER et al., supra, was unable to find a distinguishing word theme among text about genes in this cluster. Therefore, the words in **Table 2** may not reflect a common theme for genes in this cluster.

Figs. 5 and **6** give examples of results for clusters generated by the Clustering Module **(124)** itself, after estimating mRNA synthesis rates. The clusters correspond to the ones whose Keywords were given in Tables 3 and 4, respectively. The results shown in Fig. 5 were obtained with microarray data described in IYER et al., supra, downloaded over the Internet from *genome-www.stanford.edu*, with no noise added by the Clustering Module **(124)** before processing. The results shown in Fig. 6 were

generated with the same microarray data, except that noise with a coefficient of variation of 30% was added by the Clustering Module **(124)** before processing. The figure of merit for the cluster shown in Fig. 5 is 0.240 (difference between means of the distributions corresponding to original and randomized accession number assignments). The cluster obtained after adding noise (30% coefficient of variation) to the microarray data contained 69.7% of the microarray spots present in the cluster obtained without adding noise. As shown in Fig. 6, addition of the noise caused the figure of merit to drop to 0.132. The fact that this figure of merit is still significantly greater than zero, and the fact that many of the top Keywords remain associated with the cluster even after adding noise to the original microarray data (compare Table 3 with Table 4) indicates that the results shown in Fig. 5 are robust.

Figs. 7 and **8** provide another example of the operation of our method for estimating mRNA synthesis rates from data concerning total mRNA levels. The example involves simulated data representing 100 genes. The data consists of 10 sets (clusters) of 10 genes each, with each gene in a set given the same mRNA synthesis function. The synthesis function for each cluster was a triangular spike, with the same spike height and width for each cluster, but with the time of the spike's maximum different for each cluster. Each gene in a set was given a different degradation function. The degradation function used was as follows -- each gene had a different, but constant, degradation rate. By convolving the synthesis and degradation functions, then sampling at 12 time points, we obtain the 100 total mRNA kinetic functions shown in Fig. 7. To estimate the synthesis rates from the data shown in Fig. 7, we applied the algorithm described in the Preferred Embodiments section. The result of

the first iteration of estimating the synthesis functions is shown in Fig. 8, which exhibits 10 clusters, correctly corresponding to the synthesis functions that had been used to generate the total mRNA functions shown in Fig. 7. The centroid of each of these clusters is close, but not identical to, the actual synthesis functions, because the simulated data were sampled at only 12 time points. Errors in the estimated synthesis rates are most noticeable for the cluster that peaks at 20 hours, because no samples were taken after 24 hours.The example therefore illustrates that the method can correctly cluster genes on the basis of their estimated mRNA synthesis rates, even if the estimated rates themselves contain uncertainties due to sampling of the mRNA kinetics at only a limited number of time points.

EXAMPLES USING THE ALTERNATE EMBODIMENTS

The example that is described below makes use of the same data that were used in connection with Figs. 3 and 4, which illustrated the preferred embodiment. As described below, analyzing these data with the alternate embodiment of the method gives results that agree with the analysis using the preferred embodiment.

The example makes use of microarray data that are described in IYER et al., supra, which are available at the Web site *genome-www.stanford.edu.* They are from an experiment in which quiescent human diploid fibroblasts were stimulated to proliferate by increasing the concentration of fetal bovine serum in their growth medium. At 12 time points after addition of the serum, samples of mRNA were collected and used to prepare cDNA for hybridizing to an array. Five hundred-seventeen genes on the array were observed to show significant up- or down-

regulation, as defined in IYER et al., supra. The microarray data corresponding to these 517 genes were clustered using a hierarchical clustering algorithm [EISEN et al., supra], resulting in 10 clusters, labeled A through J in fig. 2 of IYER et al., supra. Accession numbers for all the genes represented as spots on the microarray are available at the Web site given above.

Analysis of these data made use of information extracted from Build 96 of the Unigene database, which was contained in the file Hs.data.Z, downloaded from the Web site *ftp.ncbi.nlm.nih.gov/pub/schuler/Unigene.* The analysis also made use of information extracted from the Locus Link database (version September 10, 1999), which was downloaded as the file LL_templ at the Web site *ftp.ncbi.nlm.nih.gov/refseq/LocusLink* . Processing of these data to associate the accession numbers with Omim numbers proceeded as described in the Preferred Embodiment, through pointers from accession numbers to UniGene numbers to LocusLink numbers to Omim numbers. Among the 517 accession numbers in the test data, 237 of them could be associated with Omim numbers. Most of the others were unidentified ESTs, which were excluded from further analysis by associating them with Omin number "0".

The UID Identification Module **(120)** in **Fig. 2** then downloaded Omim Web pages corresponding to those Omim numbers and extracted from them Unique IDentifier numbers ("uids"), as also described in the Preferred Embodiment. A total of 19,643 uids were associated with 237 Omim numbers, 1944 of which were associated with more than one Omim number.

The Clustering Module **(124)** in **Fig. 2** then clustered the 517 accession numbers using an externally available clustering scheme, which is the one shown in Fig. 2 of IYER et al (1999). The clusters, associated with the accession numbers of the clustered genes, were obtained at the Web site http://genome-www.stanford.edu. For use in the example, the clusters that IYER et al (1999) had labeled A through J were relabeled 1 through 10, and accession numbers that had not been associated with a cluster were put into a cluster "zero".

For every possible pair of Omim numbers in each of the subdirectories \0, \1, ...\Cmax, the Citation Analysis Module **(200)** went down the list of uid numbers associated with one member of the pair. As it proceeds, it checks to see whether that uid number is also found in the list of uid numbers associated with the other member of the Omim pair. The total number of matches that it found was then divided by the total number of uid numbers associated with one or the other Omim number, whichever of the two have the shorter list. This ratio is defined as the 'coupling strength' between the two Omim numbers.

These coupling strength values, which are associated with each possible pair of Omim numbers (within a subdirectory \0, \1, ..., or \Cmax) were then stored in the section Coupling Strength Data **(202)** in the Data Repository **(138)**. When all the coupling strengths were calculated for the pairs of Omin numbers in a subdirectory corresponding to a cluster, these coupling strengths were then averaged to provide an average coupling strength for that cluster, which is also stored in the section Coupling Strength Data **(202)** in the Data Repository **(138)**.

When all the average coupling strength values had been calculated , the Process Control Module **(116)** passed control to the Cluster Randomization Module **(132)**. It randomized the assignment of accession numbers to clusters, using the same method that was described in connection with the preferred embodiment. The Process Control Module **(116)** then passes control back to the Citation Analysis Module **(200)**. The process of calculating an average coupling strength for each of the clusters was then repeated as described above, namely, the association of Omim numbers to the clusters based on their (randomized) accession number composition, the estimation of coupling strength for each pair of Omim numbers in each cluster, and the calculation of an average coupling strength for each cluster. Those results are also stored in the Coupling Strength Data section **(202)** of the Data Repository **(138)** for subsequent use. The entire process of accession number randomization and coupling strength calculation is repeated the number of times that was specificed as an option by the User (default number of times=100).

The Process Control Module **(116)** then passes control to the Data Summarization Module **(134)**. For each cluster, the average coupling strengths for the randomized clusters are sorted in ascending order. Then, as an index of the sigificance of the average coupling strength that had been obtained before randomizing the clusters, the Data Summarization Module **(134)** determines the percentage of times that the observed coupling strength for each cluster was larger than those obtained with the randomized clusters. For example, if exactly half of the average coupling scores obtained by randomizing the cluster was less than the average coupling score before randomization, the percentage coupling score for that cluster would be 50%.

For the data in the example, Cluster D had a percentage coupling score of 82% suggesting that many genes in this cluster are functionally related. Cluster B, on the other hand, had a percentage coupling strength of only 25%. These results are in agreement with the results shown in Figs. 3 and 4, which also indicated that the accession numbers (genes) associated with Cluster D are functionally related to one another, whereas the accession numbers (genes) associated with cluster B are not.

CONCLUSION AND SCOPE OF INVENTION

While the above description contains many specifications, these should not be construed as limitations on the scope of the invention, but rather as an exemplification of preferred embodiments thereof.